

UNIVERSIDADE PAULISTA – UNIP



CURSO SUPERIOR DE TECNOLOGIA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

TEMA: Projeto Integrado Multidisciplinar - PIM 1º/4º Semestre

Alunos:

GUSTAVO MELO DE LIMA PEREIRA - R.A.:F363JC0

GABRIEL CALEBE TEIXEIRA BRANDÃO - R.A.:H688671

LETICIA MOCCI DEZANETE - R.A.:H765GB8

RAFAEL HENRIQUE JUBILATO BATISTA - R.A.: H70CJG2

RICKSON TADEU CANDIDO PEDREIRA - R.A.: H520BA0

PEDRO H. DOS REIS CASSIANO DA SILVA – R.A: H765CC7

PEDRO SANTANA FRANÇA – R.A: R854BD3

Coordenador:

PROF. ME. JONATHAN ROGERI

SÃO JOSÉ DO RIO PRETO

2025

Alunos:

GUSTAVO MELO DE LIMA PEREIRA - R.A.:F363JC0

GABRIEL CALEBE TEIXEIRA BRANDÃO - R.A.:H688671

LETICIA MOCCI DEZANETE - R.A.:H765GB8

RAFAEL HENRIQUE JUBILATO BATISTA - R.A.: H70CJG2

RICKSON TADEU CANDIDO PEDREIRA - R.A.: H520BA0

PEDRO H. DOS REIS CASSIANO DA SILVA – R.A: H765CC7

PEDRO SANTANA FRANÇA – R.A: R854BD3

TEMA

- Projeto Integrado Multidisciplinar - PIM 1º/4º Semestre
- Projeto Integrado Multidisciplinar para o 1º semestre do Curso

Graduação:

- Superior de Tecnologia - Análise e Desenvolvimento de Sistemas apresentado à Universidade Paulista – UNIP.

Coordenador:

PROF. ME. JONATHAN ROGERI

RESUMO

O presente trabalho descreve a concepção e implementação de um Ambiente Didático voltado à Aprendizagem de Python, desenvolvido no âmbito do Projeto Integrado Multidisciplinar (PIM) do curso de Análise e Desenvolvimento de Sistemas. Nossa proposta visa facilitar o processo de familiarização com a linguagem Python para estudantes em fase inicial de aprendizado, mediante uma plataforma que privilegia a experimentação prática e a interatividade. A ferramenta incorpora funcionalidades essenciais como autenticação de usuários, materiais didáticos estruturados, ambiente sandbox para experimentação de código, mecanismos de avaliação e painéis gráficos para acompanhamento do progresso individual. Na implementação técnica, recorreremos ao ecossistema Python com foco em Streamlit para a interface, SQLAlchemy para persistência de dados e diversas bibliotecas complementares. O conteúdo programático contempla os elementos fundamentais da lógica de programação, abrangendo desde o tratamento de variáveis até estruturas mais complexas como condicionais, iterações e definição de funções. Os testes realizados evidenciaram a eficácia da solução proposta tanto na transmissão de conceitos introdutórios de programação quanto no estímulo ao desenvolvimento do pensamento algorítmico dos estudantes.

Palavras-chave: Python; programação; lógica de programação; ensino de programação.

ABSTRACT

This paper describes the design and implementation of a Python Learning Environment, developed as part of the Integrated Multidisciplinary Project (PIM) of the Systems Analysis and Development course. Our proposal aims to facilitate the process of familiarization with the Python language for students in the early stages of learning, using a platform that *favours* practical experimentation and interactivity. The tool incorporates essential features such as user authentication, structured teaching materials, a sandbox environment for code experimentation, assessment mechanisms and graphical dashboards for monitoring individual progress. In the technical implementation, we used the Python ecosystem with a focus on *Streamlit* for the interface, *SQLAlchemy* for data persistence and various complementary libraries. The syllabus covers the fundamental elements of programming logic, from the treatment of variables to more complex structures such as conditionals, iterations and the definition of functions. The tests carried out showed that the proposed solution was effective both in transmitting introductory programming concepts and in stimulating the development of students' algorithmic thinking.

Keywords: Python; programming; programming logic; teaching programming.

Sumário

INTRODUÇÃO	5
DESENVOLVIMENTO	5
Propósito da Ferramenta	6
Abordagem de Construção	6
Estratégias para menor consumo de Energia	7
ESPECIFICAÇÕES TÉCNICAS	7
Requisitos Operacionais	7
Requisitos de Qualidade	7
Interações com o Sistema	8
ESTRUTURAS E DESENVOLVIMENTO	8
Organização Técnica	8
Estruturação de Dados	8
Módulos Principais	9
Análise Contextual de Problemas	10
Material Didático	10
Sistema de Verificação de Aprendizado	10
Representação Gráfica de Resultados	11
Interface do Estudante	11
VERIFICAÇÃO E VALIDAÇÃO	12
Testes de Funcionalidade	12
Testes de Proteção	12
Testes de Experiência	12
APÊNDICE	13
CONCLUSÃO	13
Desenvolvimentos Futuros	13
REFERÊNCIAS	14

INTRODUÇÃO

Nos dias atuais, marcados por uma intensa evolução tecnológica e digitalização dos processos, o aprendizado de linguagens de programação tornou-se cada vez mais valorizado no mercado de trabalho. Aprender a programar desde os níveis mais básicos é uma forma eficiente de desenvolver o pensamento lógico, a criatividade e a capacidade de solucionar problemas. Nesse contexto, a linguagem Python tem se destacado como uma das melhores opções para iniciantes, por sua sintaxe clara, simples e próxima da linguagem humana, além de ser uma linguagem de alto nível com ampla aplicação em áreas como ciência de dados, automação, desenvolvimento web e inteligência artificial. (PYTHON SOFTWARE FOUNDATION, 2025)

Este documento apresenta o Ambiente Didático para Aprendizagem de Python, uma ferramenta desenvolvida como requisito do Projeto Integrado Multidisciplinar I (PIM) do curso de Análise e Desenvolvimento de Sistemas. O sistema foi idealizado com o objetivo de auxiliar iniciantes na compreensão de fundamentos de programação e matemática, por meio de uma abordagem prática e interativa. Para isso, foram utilizadas tecnologias modernas como Python (PYTHON SOFTWARE FOUNDATION, 2025) para o núcleo de processamento, *Streamlit* (STREAMLIT, 2025) para a construção da interface web, *SQLAlchemy* (SQLALCHEMY, 2025) para o gerenciamento do banco de dados, além de bibliotecas auxiliares para funcionalidades específicas, como *Matplotlib* (MATPLOTLIB, 2025) e *bcrypt* (BCRYPT, 2025).

Os programas elaborados nesse ambiente didático focam em conceitos essenciais da lógica de programação, como o uso de variáveis, operadores, estruturas condicionais (*if/else*), laços de repetição (*for/while*), entrada e saída de dados, e a criação de funções. Através desses elementos, é possível desenvolver aplicações simples, porém funcionais, como calculadoras, jogos de adivinhação, sistemas de verificação e automatizações básicas. Além disso, a ampla documentação e a comunidade ativa em torno do Python oferecem suporte contínuo ao aprendizado, estimulando o desenvolvimento de projetos práticos que facilitam a inserção dos alunos nas diversas áreas da tecnologia (PYTHON SOFTWARE FOUNDATION, 2025).

DESENVOLVIMENTO

O aprendizado de programação requer o conhecimento de conceitos lógicos e estruturais que são frequente à maioria das linguagens, como variáveis, operadores, estruturas de decisão, laços de repetição e funções. Nos dias atuais, a linguagem Python é destacada por sua sintaxe de facilitar e deixar as coisas com mais clareza, o que ajuda no entendimento desses elementos, mesmo para quem nunca utilizou uma programação. Uma das primeiras ideias ensinadas aos iniciantes é a criação de variáveis, que facilitam armazenar e manipular dados. Em Python, esse serviço é feito de maneira direta, sem a obrigação de declarar o tipo da variável, o que torna o sistema menos complexo. Em seguida, os alunos tentam solicitar entradas do usuário utilizando a função `input()`, e expor saídas com `print()`. Essa relação simples permite que os

primeiros programas realizem tarefas úteis, como calcular a idade de uma pessoa, somar dois números ou converter temperaturas. As estruturas instruídas, em termos de repetição, como *for* e *while*, utilizam o código executando instruções múltiplas vezes de forma automatizada. Esses atendimentos são aplicados em pequenos projetos, como jogos de adivinhação, validadores de senha, ou verificadores de número primo. Essas atividades ajudam o conhecimento por meio da prática e do raciocínio lógico. Ademais, o uso de funções em Python, determinada pela palavra-chave *def*, influencia a reutilização de código e a organização do programa, características essenciais para a programação estruturada e, no futuro, para a instrução dos objetos. Futuramente, os iniciantes também desenvolvem a importar bibliotecas nativas ou externas, como *math*, *random* e *datetime*, que aumenta ainda mais as capacidades de seus projetos. desenvolver em programa em Python também desenvolve habilidades que vão além da tecnologia, com solução de problemas, pensamento analítico e inovador. A linguagem é utilizada nas áreas de automação de tarefas, desenvolvimento de sites, análise de dados, inteligência artificial, entre outras, o que aumenta a importância dos alunos ao conhecerem sua função nos últimos tempos. Vale revelar que a grande população de desenvolvedores e a múltipla documentação oficial são fatos que ajudam para um aprendizado mais acessível, oferecendo que qualquer pessoa com acesso à internet possa iniciar seus estudos em programação de forma gratuita e autônoma.

Propósito da Ferramenta

- Criar um ambiente virtual dedicado ao aprendizado dos fundamentos de Python
- Estabelecer conexões entre teorias matemáticas e sua codificação prática
- Disponibilizar espaço protegido para testes e experimentações de código
- Fornecer retorno pedagógico imediato sobre atividades realizadas
- Estruturar mecanismos de verificação da absorção do conhecimento

Abordagem de Construção

O projeto foi estruturado seguindo uma metodologia progressiva, contemplando as etapas:

1. Levantamento de necessidades e escopo funcional
2. Construção do esqueleto básico do aplicativo
3. Desenvolvimento do sistema de gerenciamento de usuários
4. Elaboração dos materiais didáticos interativos
5. Implementação do módulo avaliativo
6. Verificação e aprimoramento das funcionalidades

Estratégias para menor consumo de Energia

Embora o consumo de energia elétrica em aplicações web seja relativamente baixo, o projeto adotou boas práticas para otimizar recursos, contribuindo indiretamente para o uso consciente de energia. Entre as práticas aplicadas destacam-se:

1. Redução de processamento desnecessário, com código otimizado e modularizado
2. Execução local dos testes, evitando uso excessivo de servidores externos
3. Uso de Streamlit, que permite construção de aplicações leves e com baixo uso de CPU
4. Evitar loops infinitos ou mal condicionados
5. Armazenamento de dados eficiente com SQLite, evitando consultas excessivas

ESPECIFICAÇÕES TÉCNICAS

Requisitos Operacionais

1. A ferramenta deve possibilitar a criação de contas individuais
2. O sistema deve validar credenciais de acesso dos estudantes
3. O sistema utilizado no projeto foi Windows 11 Pro
4. O aplicativo deve disponibilizar conteúdo educativo sobre manipulação de textos em Python
5. A plataforma deve apresentar material sobre operações matemáticas em Python
6. O ambiente deve possibilitar a execução controlada de scripts Python
7. O sistema deve reconhecer e explicar erros típicos de iniciantes
8. A plataforma deve possibilitar a verificação do aprendizado através de questões
9. O aplicativo deve exibir métricas visuais de desempenho após avaliações

Requisitos de Qualidade

- O aplicativo deve ser utilizável através de navegadores web comuns
- As credenciais de acesso devem ser armazenadas com proteção criptográfica
- A execução de códigos deve ocorrer em ambiente isolado para evitar vulnerabilidades
- A interface de uso deve ser fluida e autoexplicativa

- As interações devem gerar respostas imediatas e contextualizadas

Interações com o Sistema

Os principais fluxos de interação identificados incluem:

- Processo de registro de novo estudante
- Autenticação no ambiente educacional
- Navegação pelos materiais didáticos
- Experimentação de códigos Python
- Realização de exercícios práticos
- Participação em testes avaliativos
- Análise de resultados obtidos

ESTRUTURAS E DESENVOLVIMENTO

Organização Técnica

A ferramenta foi implementada seguindo uma estrutura de aplicativo web com as seguintes camadas:

- Camada de apresentação: Construída com Streamlit para interface responsiva
- Camada lógica: Desenvolvida em Python para processamento de regras
- Camada de persistência: Implementada com SQLite e SQLAlchemy
- Componentes auxiliares: Matplotlib para visualizações e bcrypt para segurança

Estruturação de Dados

O aplicativo utiliza uma modelagem simplificada, com foco na entidade principal:

Código 1 Estrutura do Cadastro de Estudante

1. *class User (Base):*
2. *__tablename__ = 'users'*
3. *id = Column (Integer, primary_key=True, autoincrement=True)*
4. *email = Column (String, unique=True, nullable=False)*

5. *senha = Column (String, nullable=False)*

6. *nome = Column (String, nullable=False)*

Módulos Principais

O módulo de controle de acesso implementa técnicas de proteção utilizando *bcrypt*:

Código 2: Métodos de Gerenciamento de Acesso

```
1. def registro_user (email, senha, nome):  
2. hashed_password = bcrypt.hashpw(senha.encode('utf-8'), bcrypt.  
3. gensalt()).decode('utf-8')  
4. novu_user = User(nome=nome, email=email, senha=hashed_password) 4 try:  
5. session.add(novo_user)  
6. session.commit()  
7. return True  
8. except Exception as e: 9 session.rollback()  
9. print(f"Erro: {e}")  
10. return False  
11. 12  
12. def login_user(email, senha):  
13. user = session.query(User).filter_by(email=email).first()  
14. if user and bcrypt.checkpw(senha.encode('utf-8'), user.senha.  
15. encode('utf-8')):  
16. return user.nome  
17. return None
```

Análise Contextual de Problemas

O módulo de análise de erros oferece explicações adaptadas ao nível de iniciantes:

Código 3: Identificação e Explicação de Erros

```
1. def interpretar_erro(e):
2.     if isinstance(e, SyntaxError):
3.         return "Erro de sintaxe! Isso pode ser causado por parênteses ou aspas não fechadas."
4.     elif isinstance(e, NameError):
5.         return "Você usou uma variável ou função não definida. Verifique a grafia."
6.     elif isinstance(e, ZeroDivisionError):
7.         return "Você tentou dividir por zero. Isso não permitido!"
8.     elif isinstance(e, TypeError):
9.         return "Erro de tipo: talvez você esteja tentando somar uma string com um número."
10. else:
11.     return f"Erro desconhecido: {str(e)}"
```

Material Didático

A ferramenta apresenta conteúdos educativos sobre temas fundamentais:

- Manipulação de textos: conceitos, operações e exemplos
- Cálculos matemáticos: operadores básicos e avançados, precedência
- Comandos de saída: exibição de resultados, formatação

Sistema de Verificação de Aprendizado

O módulo de avaliação inclui questões de múltipla escolha sobre os temas abordados:

Código 4: Organização das Questões Avaliativas

```
1. questoes = [  
2. {  
3. "pergunta": "O que uma string em Python?",  
4. "opcoes": ["a) Um número inteiro", "b) Um tipo de dados que representa textos",  
5. "c) Um tipo de dados usado para cálculos", "d) Um erro no código"],  
6. "resposta_correta": "b) Um tipo de dados que representa  
7. textos"  
8. },  
9. # Outras quest es...  
10.]
```

Representação Gráfica de Resultados

A ferramenta utiliza Mplotlib para visualização dos resultados avaliativos:

Código 5: Geração de Gráficos de Desempenho

```
1. desempenho = [acertos, len(questoes) - acertos]  
2. labels = ['Acertos', 'Erros']  
3. fig, ax = plt.subplots()  
4. ax.pie(desempenho, labels=labels, autopct='%1.1f%%', startangle=90)  
5. ax.axis('equal')  
6. st.pyplot(fig)
```

Interface do Estudante

Tela inicial de acesso, a tela inicial apresenta opções para ingresso no sistema, seja através de conta existente ou criação de novo perfil. Para cadastro é necessário nome, *email* e senha. Para acesso, bastam as credencias *email* e senha.

O ambiente de estudo disponibiliza blocos expansíveis com o conteúdo didático. Cada bloco contém:

- Explicações conceituais sobre o tema
- Demonstrações práticas e exemplos
- Espaço para experimentação prática do estudante

A seção de avaliação contém questões de múltipla escolha relacionadas aos temas estudados. Ao finalizar, o estudante visualiza um resumo gráfico do seu desempenho.

VERIFICAÇÃO E VALIDAÇÃO

Testes de Funcionalidade

Os testes de funcionalidade verificam o comportamento correto dos seguintes componentes:

- Processos de registro e autenticação de estudantes
- Exibição adequada dos materiais didáticos
- Funcionamento do ambiente de execução de código
- Identificação correta de erros comuns
- Mecanismo de avaliação e exibição de resultados

Testes de Proteção

Os testes de proteção verificaram aspectos como:

- Segurança no armazenamento de senhas
- Execução isolada de código Python
- Validação apropriada das entradas dos usuários

Testes de Experiência

Os testes de experiência do usuário avaliaram:

- Facilidade de utilização das diferentes seções
- Clareza na apresentação das informações
- Adequação das respostas e retornos ao estudante

APÊNDICE

A seguir, disponibilizamos os principais materiais de apoio relacionados ao projeto, incluindo o repositório com o código-fonte e a demonstração em vídeo da aplicação desenvolvida:

- Repositório GitHub: <https://github.com/RFL370z/PIM-1>
- Vídeo da aplicação:



Ficha de controle
do PIM - REUNIOES.

- Ficha de Controle das Reuniões:

CONCLUSÃO

O Ambiente Didático para Aprendizagem de Python atinge seu objetivo principal de oferecer uma ferramenta educacional interativa para iniciantes em programação e matemática aplicada. A solução integra funcionalidades de controle de acesso, apresentação de conteúdo, experimentação prática e avaliação, fornecendo uma experiência educacional completa.

A utilização da biblioteca Streamlit permitiu a criação de uma interface web intuitiva e acessível, enquanto o SQLAlchemy simplificou o gerenciamento dos dados de usuários. A incorporação de funcionalidades como análise contextualizada de erros e visualização gráfica de resultados enriquece significativamente a experiência educacional.

Desenvolvimentos Futuros

Potenciais aprimoramentos para a ferramenta incluem:

- Expansão do material didático com tópicos intermediários e avançados
- Introdução de diferentes níveis de complexidade adaptados ao perfil do estudante
- Incorporação de elementos de gamificação para aumentar o engajamento
- Implementação de acompanhamento progressivo da evolução do aprendiz
- Adição de recursos colaborativos entre estudantes

REFERÊNCIAS

BCRYPT: A library to hash and check passwords using bcrypt. Disponível em: <https://pypi.org/project/bcrypt/>. Acesso em: 10 mai. 2025.

MATPLOTLIB: Visualization with Python. Disponível em: <https://matplotlib.org/>. Acesso em: 10 mai. 2025.

PYTHON SOFTWARE FOUNDATION: Python 3.10 Documentation. Disponível em: www.python.org. Acesso em: 10 mai. 2025.

SQLALCHEMY: SQLAlchemy Documentation. Disponível em: <https://docs.sqlalchemy.org/>. Acesso em: 10 mai. 2025.

STREAMLIT: Streamlit Docs. Disponível em: <https://docs.streamlit.io>. Acesso em: 10 mai. 2025.